Journal of information and
communication convergence engineering
http://jicce.org

J. Inf. Commun. Converg. Eng. 22(2): 139-144, Jun. 2024

Regular paper

Comparison of Fall Detection Systems Based on YOLOPose and Long Short-Term Memory

Seung Su Jeong[1], **Nam Ho Kim**[2]*, **and Yun Seop Yu**[1]*, ***Member, KIICE***

[1]ICT & Robotics Engineering and IITC, Hankyong National University, Anseong 17579, Republic of Korea
[2]Bundang Convergence Technology Campus of Korea Polytechnic, Seongnam 13590, Republic of Korea

Abstract

In this study, four types of fall detection systems – designed with YOLOPose, principal component analysis (PCA), convolutional neural network (CNN), and long short-term memory (LSTM) architectures – were developed and compared in the detection of everyday falls. The experimental dataset encompassed seven types of activities: walking, lying, jumping, jumping in activities of daily living, falling backward, falling forward, and falling sideways. Keypoints extracted from YOLOPose were entered into the following architectures: RAW-LSTM, PCA-LSTM, RAW-PCA-LSTM, and PCA-CNN-LSTM. For the PCA architectures, the reduced input size stemming from a dimensionality reduction enhanced the operational efficiency in terms of computational time and memory at the cost of decreased accuracy. In contrast, the addition of a CNN resulted in higher complexity and lower accuracy. The RAW-LSTM architecture, which did not include either PCA or CNN, had the least number of parameters, which resulted in the best computational time and memory while also achieving the highest accuracy.

Index Terms: Fall detection, The elderly, Long short-term memory (LSTM), Principal component analysis (PCA), Convolutional neural network (CNN)

I. INTRODUCTION

According to the World Health Organization (WHO), falls are the second leading cause of unintended, unexpected, or accidental death [1]. Furthermore, falls often result in serious injuries such as femur neck fractures, neurological damage, and skin burns [2]. In the process of physical activities, falls can be detected using sensors, images, or videos. Advances in computer vision and network technologies have made it possible to quickly detect falls and transmit relevant information, thereby enabling rapid detection and response. In addition, a system was developed to track and monitor users in real time, allowing feedback to be sent to medical professionals [3]. Users can be monitored via wearable sensors, such as gyroscopes and accelerometers [4-7]; visual sensors, such as cameras [8-12]; or ambient sensors, such as active infrared sensors, RFID, ultrasonic sensors, radar, and microphones [13-16].

In video-based approaches, motion information is acquired as features extracted from image frames captured by a camera. Human movements can be visually classified using systems such as convolutional neural networks (CNNs), human position estimation (HPE), object detection, and optical flow. A fall detection system based on principal component analysis (PCA) has previously been developed using optical flow [17,18], which refers to the visible movement of light patterns in a video. Optical flow reflects changes in lighting that stem from motion within an image and connects lines over time. Specifically, the flow of light is calculated using the Luca–Kanade algorithm to determine movement. How-

Received 19 February 2024, **Revised** 11 April 2024, **Accepted** 13 April 2024
***Corresponding Author** Yun Seop Yu[1] and Nam Ho Kim[2] (E-mail: ysyu@hknu.ac.kr, namo@kopo.ac.kr, Tel: +82-31-670-5293, +82-31-696-8851)
[1]ICT & Robotics Engineering and IITC, Hankyong National University, Anseong 17579, Republic of Korea
[2]Bundang Convergence Technology Campus of Korea Polytechnic, Seongnam 13590, Republic of Korea

Open Access **https://doi.org/10.56977/jicce.2024.22.2.139** print ISSN: 2234-8255 online ISSN: 2234-8883

ever, this method does not differentiate between the motion of people and that of objects. Furthermore, this approach is vulnerable to fast-moving objects, and even if no motion has occurred, the system may misinterpret changes in contrast caused by camera noise as motion. To compensate for these shortcomings, HPE techniques [8-12] have been developed to identify people through deep learning networks, such as a CNNs. These techniques are designed to track identified persons by locating key points, such as those corresponding to the human skeleton. Among recently developed HPE methods, YOLOPose can simultaneously identify multiple people within an image or a video, enabling relatively accurate real-time processing [19,20]. The YOLOPose model is a kinetic model that predicts 17 keypoints in a 2D human image. In fall detection systems, data collected over time are used to effectively identify occurrences of falling [9]. Recurrent neural networks (RNNs) [21], long short-term memory (LSTM) [22], and gate recurrent units (GRUs) [10] have all been employed for fall detection. In particular, the LSTM and GRU architectures, as well as CNN-LSTM hybrid models, have recently been utilized to solve the vanishing gradient problem inherent to RNNs [12,15]. The following fall detection architectures have been developed for HPE methods:

1. Architecture 1: Keypoints corresponding to the skeleton are measured via kinetic sensors, such as Microsoft Kinect RGBD [12] or HPE methods [22], and directly passed into an LSTM network.
2. Architecture 2: Keypoints extracted via HPE are passed to a CNN or CNN-LSTM network. In the latter, convolutional layers are used to extract input data features while the LSTM network supports sequence prediction [9,10,12,23,24].
3. Architecture 3: Geometric information pertaining to the human body is calculated to reduce the input dimensionality, and the extracted features are passed to an LSTM model and support vector machine (SVM) to classify falls and activities of daily living (ADL) [25].
4. Architecture 4: Data preprocessed via dimensional reduction, such as extracted keypoints, are passed to a CNN or CNN-LSTM network.

To our knowledge, a comparative evaluation of the four architectures in terms of accuracy, computation time, and memory efficiency has never been conducted previously.

This article presents four fall detection algorithms that classify ADL and falls using keypoint data representing four types of ADL and three types of falls extracted by YOLOPose. In a comparative evaluation, the accuracy of inference, computation time, and memory efficiency were calculated for all four algorithms. Section 2 describes the subjects and datasets used throughout this study, as well as the proposed methods, and Section 3 presents the results of the comparative evaluation. Finally, Section 4 concludes the study.

II. MATERIALS and METHODS

A. Subjects and dataset

A dataset comprising fall and ADL occurrences was compiled, encompassing 1190 images [18], of which 20 and 10 images for each fall and ADL were obtained from seven and three applicants, respectively. To enhance the diversity of training data, the images were augmented by horizontal reflection, for a total of 2380 images. Seven activities were included in the dataset: walking, lying, jumping, jumping in ADL, falling backward, falling forward, and falling sideways [18]. All individuals represented in the data were 20-50 years old, 160-180 cm tall, and weighing 50-85 kg. Images were recorded at 50 frames per second (FPS), with the framerate dropping to 25 FPS in the event of a fall. The video data were recorded at 10 FPS owing to the environmental and systemic variability of CCTVs, as this framerate is considered a minimal standard. Furthermore, a framerate of 10 FPS consumes very little memory, ensuring a highly efficient training process.

B. Proposed fall detection algorithms

Fig. 1 depicts four types of fall detection architectures, showing the 17 keypoints extracted from YOLOPose along with the eigenvector matrices $\mathbf{U_1}$ and $\mathbf{U_2}$ obtained from the PCA, 1DCNN, and LSTM networks. In the RAW-LSTM and RAW-CNN-LSTM architectures, keypoints were passed to the LSTM directly, whereas in the PCA-LSTM and PCA-CNN-LSTM models, keypoints were first dimensionally reduced via PCA. The YOLOPose network was used to extract 34 data points corresponding to the x and y coordinates of 17 keypoints identified within each video frame. In the PCA-enabled architectures, the center point (m_x, m_y) and slope (θ) of each keypoint were extracted. With the addition of a CNN architecture, a feature map was generated using either the feature points obtained via PCA, or the raw keypoint data. To construct the feature map, a 1D filter was used to extract feature points, with padding used to maintain the time component. The stride was set to 1, and the filter size was set to 2. No additional layers were used. Although CNNs typically extract feature points from images using a 2D filter, the data used in this study were separated into the x and y components, and 1D filters were used instead. The CNN consisted of two layers, with the first and second input values set to 64 and 128, respectively. The LSTM network was trained to classify the seven activities using three types of inputs: raw skeleton keypoint data obtained via YOLOPose, feature points extracted via PCA, and feature map data extracted by the CNN. The LSTM network comprised three layers and one dense layer. The first, second, and third input

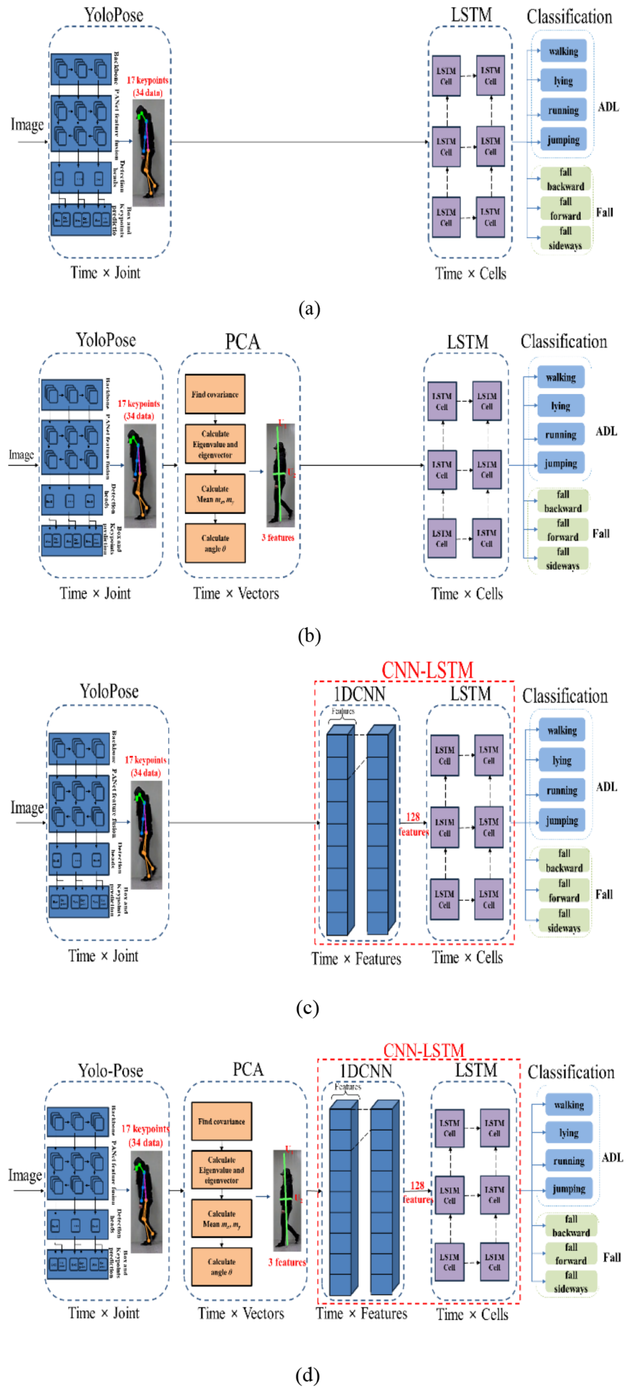


Fig. 1. Proposed fall detection architectures. (a) Raw-LSTM, (b) PCA-LSTM, (c) RAW-CNN-LSTM, and (d) PCA-CNN-LSTM.

values were set to 64, 128, and 128, respectively. The feature map was split to enable the classification of seven activities in the dense layer, with Softmax predicting the most likely behavior. Adaptive Momentum was used as the optimizer. In the hybrid CNN-LSTM architecture, data features are extracted by the CNN before being passed to the LSTM [9,10,12, 23,24]. Fig. 2 presents an overall flowchart of all four algorithms. The processed data were allocated between training and testing sets through a validation process at an 8:2 ratio [26]. To prevent overfitting, L2 regularization was applied with a lambda value of 0.065, as determined by the parameter optimization method proposed in [6]. Upon successful completion of training, each model's accuracy was determined using the validation data.

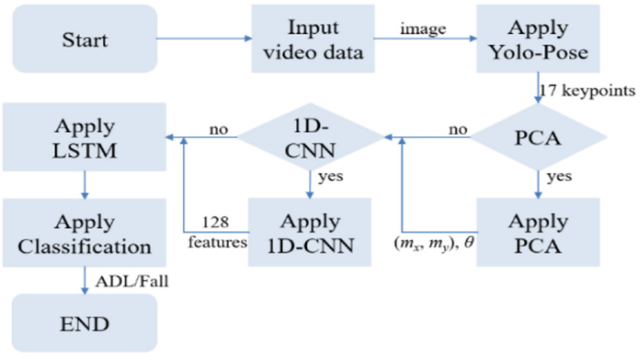


Fig. 2. Flow chart of proposed fall detection algorithms.

Min-max normalization [6] was applied to the feature points extracted via PCA, with the normalized (m_x, m_y) and θ used as indicators of speed and slope for fall detection, respectively.

C. Parameter calculation

The number of parameters in the CNN components of the hybrid models can be expressed as [27]

$$NP_{CNN} = N_{input}\ N_{fh}\ N_{fw}\ N_{kernel} + N_{kernel}, \tag{3}$$

where N_{input}, N_{fh} N_{fw}, and N_{kernel} represent the input, filter height, filter width, and kernel size, respectively. The number of parameters in the LSTM network is expressed as [28]

$$NP_{LSTM} = 4[(N_{input} + 1)\ N_{output} + N_{output}^2], \tag{4}$$

where N_{input} and N_{output} are the input and output sizes, respectively.

When PCA was employed to reduce input size, the number of parameters also decreased. This in turn decreased the algorithm complexity, thereby enhancing memory efficiency and the speed of computation. Tables 1 and 2 summarize the output shape and number of parameters used in each of the proposed architectures, with parentheses indicating parameters used after applying PCA. For example, the first LSTM layer of the RAW-LSTM architecture has input and output sizes of 34 and 64, respectively; thus, the total number of parameters is calculated as 4 × ((34 + 1) × 64 + 642) = 25344. In contrast, the first CNN layer of the PCA-CNN-LSTM architecture has a 1DCNN filter height, filter width, input

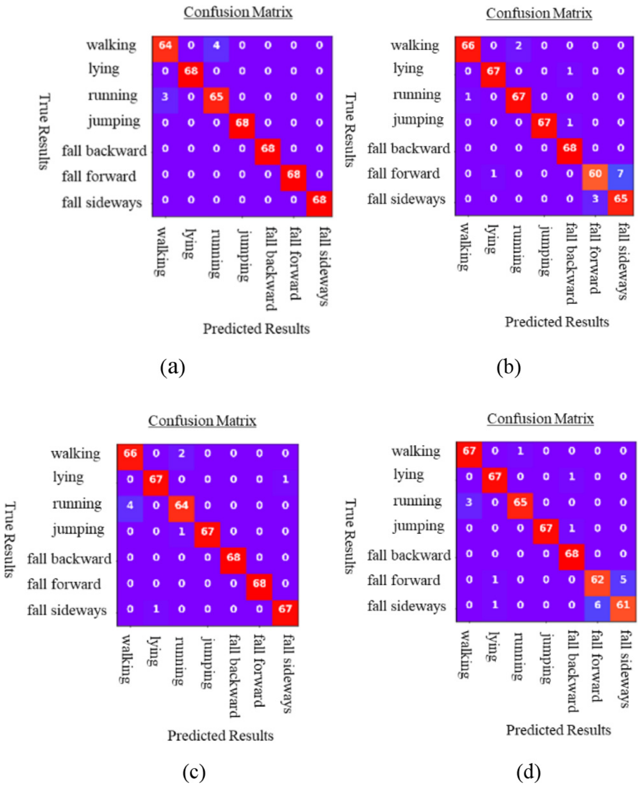
Table 1. Details of LSTM layers

Layer (Type)	Output Shape	Number of Parameters
LSTM_1 (LSTM)	(None, 50, 64)	25344 (17408)
LSTM_2 (LSTM)	(None, 50, 128)	98816
LSTM_3 (LSTM)	(None, 128)	131584
Dense_1 (Dense)	(None, 7)	903
Total number of parameters: 256,647 (248,711)		

Table 2. Details of CNN-LSTM layers

Layer (Type)	Output Shape	Number of Parameters
CONV1D_1 (CONV1D)	(None, 50, 64)	4416 (448)
CONV1D_2 (CONV1D)	(None, 50, 128)	8256
LSTM_1 (LSTM)	(None, 50, 64)	33024
LSTM_2 (LSTM)	(None, 50, 128)	98816
LSTM_3 (LSTM)	(None, 128)	131584
Dense_1 (Dense)	(None, 7)	903
Total number of parameters: 276,999 (273,031)		

size, and kernel size of 1, 2, 3, and 64, respectively; thus, the number of parameters is calculated as 3 × 1 × 2 × 64 + 64 = 448.



(a) (b)

(c) (d)

Fig. 3. Confusion matrix applying the proposed fall detection system (a) Raw-LSTM, (b) PCA-LSTM, (c) Raw-CNN-LSTM, (d) PCA-CNN-LSTM.

III. RESULTS

Fig. 3 shows the confusion matrices obtained by the four architectures using the validation data, with Table 3 listing the corresponding evaluation results. Because the data were distributed between training and validation sets at a ratio of 8:2, 1904 and 476 activities were used for training and validation, respectively. From the results, a slight decrease in accuracy can be observed following the application of PCA, which can be regarded as the cost of the enhanced computation time and memory efficiency. The validation accuracy was also slightly lower when applying the CNN, which also increased model complexity. Specifically, the RAW-LSTM architecture achieved a maximal validation accuracy of 100% while also exhibiting optimal memory use and computation time, as it requires less training parameters than the hybrid architectures.

Table 3. Validation results of fall detection of four types of proposed fall detection system

Architecture	Sensitivity [%]	Specificity [%]	Accuracy [%]
Raw-LSTM	100	100	100
PCA-LSTM	99.51	99.26	99.37
Raw-CNN-LSTM	99.51	99.63	99.58
PCA-CNN-LSTM	99.02	99.26	99.16

IV. CONCLUSION

In this study, the four types of fall detection architectures were investigated in terms of accuracy, computation time, and memory efficiency. In the case of PCA applications, the reduced input dimensionality enhanced the computation time and memory efficiency while slightly decreasing accuracy. In contrast, the addition of a 1D CNN led to both increased complexity and decreased accuracy. Overall, the RAW-LSTM architecture, which did not include PCA or CNN achieved the best accuracy, computational time, and memory efficiency.

ACKNOWLEDGEMENTS

This research was supported by the Basic Science Research Program through the NRF of Korea, funded by the Ministry of Education (NRF-2019R1F1A1060383).

REFERENCES

[1] World Health Organization. Available online: https://www.who.int/news-room/fact-sheets/detail/falls (accessed on 26 April 2021).

[2] National Health Administration, Ministry of Health and Welfare. Available online: https://www.hpa.gov.tw/Pages/Detail.aspx?nodeid

=807&pid=4326 (accessed on 9 March 2020).

[3] Y. Ren, Y. Chen, M. C. Chuah, and J. Yang, “User verification leveraging gait recognition for smartphone enabled mobile healthcare systems,” *IEEE Trans. Mobile Comput.* vol. 14, pp. 1961-1974, 2014. DOI: 10.1109/TMC.2014.2365185.

[4] A. Ramachandran and A. Karuppiah, “A Survey on Recent Advances in Wearable Fall Detection Systems,” *BioMed Research International*, vol. 2020, pp. 1-17, Jan. 2020. DOI: 10.1155/2020/2167160.

[5] E. Casilari, R. Lora-Rivera, and F. García-Lagos, “A Study on the Application of Convolutional Neural Networks to Fall Detection Evaluated with Multiple Public Datasets,” *Sensors*, vol. 20, no. 5, pp. 1466, Mar. 2020. DOI:10.3390/s20051466.

[6] S. S. Jeong, N. H. Kim, and Y. S. Yu, “Fall Detection System Based on Simple Threshold Method and Long Short-Term Memory: Comparison with Hidden Markov Model and Extraction of Optimal Parameters,” *Applied Sciences*, vol. 12, no. 21, pp. 11031, Oct. 31, 2022. DOI: 10.3390/app122111031.

[7] D. Lim, C. Park, N. H. Kim, S.-H. Kim, and Y. S. Yu, “Fall-Detection Algorithm Using 3-Axis Acceleration: Combination with Simple Threshold and Hidden Markov Model,” *Journal of Applied Mathematics*, vol. 2014, pp. 1-8, 2014. DOI:10.1155/2014/896030.

[8] X. Wang, J. Ellul, and G. Azzopardi, “Elderly Fall Detection Systems: A Literature Survey,” *Frontiers in Robotics and AI,* vol. 7, Jun. 23, 2020. DOI:10.3389/frobt.2020.00071.

[9] M. Salimi, J. J. M. Machado, and J. M. R. S. Tavares, “Using Deep Neural Networks for Human Fall Detection Based on Pose Estimation,” *Sensors*, vol. 22, no. 12, pp. 4544, Jun. 16, 2022. DOI: 10.3390/s22124544.

[10] C.-B. Lin, Z. Dong, W.-K. Kuan, and Y.-F. Huang, “A Framework for Fall Detection Based on OpenPose Skeleton and LSTM/GRU Models,” *Applied Sciences*, vol. 11, no. 1, pp. 329, Dec. 2020. DOI:10.3390/app11010329.

[11] W. Chen, Z. Jiang, H. Guo, and X. Ni, “Fall Detection Based on Key Points of Human-Skeleton Using OpenPose,” *Symmetry*, vol. 12, no. 5, pp. 744, May 2020. DOI:10.3390/sym12050744.

[12] N. Lu, Y. Wu, L. Feng, and J. Song, “Deep Learning for Fall Detection: Three-Dimensional CNN Combined with LSTM on Video Kinematic Data,” *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 1, pp. 314-323, Jan. 2019. DOI: 10.1109/JBHI.2018.2808281.

[13] F. Demrozi, G. Pravadelli, A. Bihorac, and P. Rashidi, “Human Activity Recognition Using Inertial, Physiological and Environmental Sensors: A Comprehensive Survey,” *IEEE Access*, vol. 8, pp. 210816-210836, 2020. DOI: 10.1109/ACCESS.2020.3037715.

[14] E. Kanjo, E. M. G. Younis, and C. S. Ang, “Deep learning analysis of mobile physiological, environmental and location sensor data for emotion detection,” *Information Fusion*, vol. 49, pp. 46-56, Sep. 2019. DOI: 10.1016/j.inffus.2018.09.001.

[15] J. Maitre, K. Bouchard, and S. Gaboury, “Fall Detection With UWB Radars and CNN-LSTM Architecture,” *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 4, pp. 1273-1283, Apr. 2021. DOI:10.1109/JBHI.2020.3027967.

[16] H. Fu and J. Gao, “Human Fall Detection Based on Posture Estimation and Infrared Thermography,” *IEEE Sensors Journal*, vol. 23, no. 20, pp. 24744-24751, Oct. 2023. DOI: 10.1109/JSEN.2023.3307160.

[17] C.-F. Wu and S.-K. Lin, “Fall detection for multiple pedestrians using a PCA approach to 3-D inclination,” *International Journal of Engineering Business Management*, vol. 11, pp. 184797901987897, Jan. 2019. DOI:10.1177/1847979019878971.

[18] C. Park and Y. S. Yu, “Video-based fall detection algorithm combining simple threshold method and Hidden Markov Model,” *Journal of the Korea Institute of Information and Communication Engineering*, vol. 18, no. 9, pp. 2101-2108, Sep. 2014. DOI: jkiice.2014.18.9.2101.

[19] Maji, D.; Nagori, S.; Mathew, M.; Poddar, D. YOLO-Pose: Enhancing YOLO for multi person pose estimation using object keypoint similarity loss. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, New Orleans, LA, USA, 19-21 June 2022; pp. 2637-2646, [online] Available: https://arxiv.org/abs/2204.06806 (accessed on 20 June 2022).

[20] H.-C. Nguyen, T.-H. Nguyen, R. Scherer, and V.-H. Le, “Unified End-to-End YOLOv5-HR-TCM Framework for Automatic 2D/3D Human Pose Estimation for Real-Time Applications,” *Sensors*, vol. 22, no. 14, pp. 5419, Jul. 20, 2022. DOI: 10.3390/s22145419.

[21] M. M. Hasan, M. S. Islam, and S. Abdullah, “Robust Pose-Based Human Fall Detection Using Recurrent Neural Network,” In Proceedings of 2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON). IEEE, Nov. 2019. DOI: 10.1109/RAAICON48939.2019.23.

[22] K. Adhikari, H. Bouchachia, and H. Nait-Charif, “Long short-term memory networks based fall detection using unified pose estimation,” In Proceedings of Twelfth International Conference on Machine Vision (ICMV 2019). SPIE, Jan. 2020. DOI: 10.1117/12.2556540.

[23] J. Xu, Z. He, and Y. Zhang, “CNN-LSTM Combined Network for IoT Enabled Fall Detection Applications,” *Journal of Physics*: *Conference Series*, vol. 1267, no. 1, pp. 012044, Jul. 2019. DOI: 10.1088/1742-6596/1267/1/012044.

[24] M. Li and D. Kim, “Classification in Different Genera by Cytochrome Oxidase Subunit I Gene Using CNN-LSTM Hybrid Model,” *Journal of information and communication convergence engineering*, vol. 21, no. 2, pp. 159-166, Jun. 2023. DOI: 10.56977/jicce.2023.21.2.159.

[25] B. D. Romaissa, O. Mourad, N. Brahim, and B. Yazid, “Vision-Based Fall Detection Using Body Geometry,” In Proceedings of Pattern Recognition. ICPR International Workshops and Challenges, pp. 170-185, 2021. DOI:10.1007/978-3-030-68799-1_13.

[26] F. Burden and D. Winkler, “Bayesian Regularization of Neural Networks,” *Methods in Molecular Biology^{TM}*. Humana Press, pp. 23-42, 2008. DOI:10.1007/978-1-60327-101-1_3.

[27] A. Borji, “Enhancing sensor resolution improves CNN accuracy given the same number of parameters or FLOPS.” arXiv, 2021. DOI: 10.48550/arXiv.2103.05251.

[28] W. Song, C. Gao, Y. Zhao, and Y. Zhao, “A Time Series Data Filling Method Based on LSTM—Taking the Stem Moisture as an Example,” *Sensors*, vol. 20, no. 18, pp. 5045, Sep. 2020. DOI: 10.3390/s20185045.

Seung Su Jeong
received his BS from the Department of Electrical, Electronic, and Control Engineering, Hankyong National University, Anseong, Republic of Korea, in 2022. He is currently an MS student at the Department of ICT and Robotics Engineering, Hankyong National University. His current research is focused on the compact modeling of next-generation devices based on deep learning applications.

Nam Ho Kim
received his BS and MS from the Department of Electronics Engineering, Korea University, Seoul, Republic of Korea, in 1996 and 1998, respectively. He received his PhD in Signal Processing at the Graduate School of Biotechnology & Information Technology, Hankyong National University. Currently, he is an Associate Professor at the Bundang Convergence Technology Campus of Korea Polytechnic, Sengnam, Republic of Korea. His research is focused on machine learning, deep learning, and embedded systems.

Yun Seop Yu
received his BS, MS, and PhD from the Department of Electronics Engineering, Korea University, Seoul, Republic of Korea, in 1995, 1997, and 2001, respectively. From 2001 to 2002, he worked as a guest researcher at the Electronics and Electrical Engineering Laboratory, NIST, Gaithersburg, Maryland, USA. From 2014 to 2015, he worked as a visiting scholar at the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, USA. He is now a full professor at the School of ICT, Robotics, and Mechanical Engineering, Hankyong National University, Anseong, Republic of Korea. His main research interests are in the fields of modeling various nanodevices for efficient circuit simulation, as well as future memory, logic, and sensor designs using these devices. He is also interested in the fabrication and characterization of various nanodevices. He has authored and coauthored 100 internationally refereed journal publications.